17th International Conference on the Applications of Computer Science
and Mathematics in Architecture and Civil Engineering
K. Gürlebeck and C. Könke (eds.)
Weimar, Germany, 12–14 July 2006

SOLVING REVITALIZATION PROBLEMS BY THE USE OF A CONSTRAINT PROGRAMMING LANGUAGE

T.M. Lömker*

* *Technische Universität Dresden*
Zellescher Weg 17, 01069 Dresden, Germany
E-mail: thorsten.loemker@tu-dresden.de

Keywords: Revitalization, Optimization, Constraint Programming, OPL.

Abstract. *This research focuses on an approach to describe principles in architectural layout planning within the domain of revitalization. With the aid of mathematical rules, which are executed by a computer, solutions to design problems are generated. Provided that "design" is in principle a combinatorial problem, i.e. a constraint-based search for an overall optimal solution of a problem, an exemplary method will be described to solve such problems in architectural layout planning. To avoid conflicts relating to theoretical subtleness, a customary approach adopted from Operations Research has been chosen in this work [1]. In this approach, design is a synonym for planning, which could be described as a systematic and methodical course of action for the analysis and solution of current or future problems. The planning task is defined as an analysis of a problem with the aim to prepare optimal decisions by the use of mathematical methods. The decision problem of a planning task is represented by an optimization model and the application of an efficient algorithm in order to aid finding one or more solutions to the problem. The basic principle underlying the approach presented herein is the understanding of design in terms of searching for solutions that fulfill specific criteria. This search is executed by the use of a constraint programming language.*

1 INTRODUCTION

Future building tasks will be focused on the examination of existent architecture. The planner is challenged to forego designing new buildings in favor of reuse and conversion of present structures.

Reuse and conversion are strategies for conserving value that regard the life cycle of a building as an integral part of planning. Their aim is to alter unused structures through no or few architectural changes in such a way that they can again be put to good use. Reuse is limited by the premise that there will be no structural changes to the buildings, whereas conversion permits such alterations. In order to present successful alternatives to new buildings, both strategies strongly depend on the architect, when beginning to plan, to come to a conclusion on whether a building can be used applying one of the strategies. This decision is made by comparing the desired state (room program) of the building with its current state. In this early stage of planning, the analysis and evaluation of existing buildings is executed in form of pre-design sketches, which show the organisational or structural changes of the floor plans should the building continue to be used.

The essential criterion for deciding to revitalize a building is the satisfaction of the room program. In order to attain this, the procedure quickly leads to revitalization solutions that change the building through massive structural modifications. However, given the premise of sustainable treatment of existing buildings, decisions for the continued use of a building should be reached by taking into account the greatest possible conservation of present structures and the least possible alterations of their architectural state. This course of action has found little consideration in previous revitalization efforts.

A hypothesis is proposed, stating that the comparison of the room program with the floor plans of a building essentially is a combinatorial problem. Under this assumption it is examined whether solutions for conversion and reuse tasks can be produced automatically by the use of optimization processes in floor plan design. These solutions shall be produced by reordering or swapping of existing areas. The objective is to obtain feasible planning solutions by means of these computer-based processes, which will serve the architect as a basis for the further editing of the plans.

2 METHODOLOGY

Designing a building is a complex task. It can be compared to composing music, writing a poem, or creating an object of art. All these activities have in common that they share artistic principles. Unfortunately, these principles make it difficult to generalize the modus operandi of the design process. Therefore the use of a mathematical description to characterize a design problem implies the following hypotheses:

- Architectural design is affected by rules.
- Rules can be used to constrain the solution space of a design problem.
- Provided that constraints and objectives are specified by the architect, computers can extend the number of feasible solutions for a design problem.

In terms of methods used in Operations Research, it is possible to classify the above mentioned assumptions as integral components of an optimization model:

- The design solution has to meet specific requirements (Constraints).

- The design has to strive for specific goals (Objectives).
- There are choices available that might meet the constraints and objectives (Design Variables).

The optimization model itself consists of a given number of variable and constant parameters, one or more objectives, as well as a fluctuating number of constraints. Each object which belongs to the model can be accessed and altered by the use of parameters. A room, for example, is an object with geometric parameters such as length, width, and height. Objects can also imply alphanumerical parameters such as their occupancy or neighborhood. Parameters are defined in the form of variables or constants, whereas variables can be used as inputs for the optimization process. Responses result from the composition of other variables. If a variable is changed during the optimization process, dependent variables will be changed as well. Inputs and Responses are often named Optimization Variables. These variables form the basis of constraints and objective functions. Both must be functions of one or more optimization variables [2]. Within an architectural problem domain, a response variable could be the area occupied by a specific room. Through multiplication of two input parameters (width and length) a response variable would be rendered. It is of primary interest that suchlike parameters generate serious problems for the optimization process due to their non-linear form. Once the design problem is stated in form of design variables, constraints, and objectives, the parameters will be passed to the optimization engine which tries to find a feasible solution to the problem. The model described above is a general model that can be applied to many problem domains beyond architecture. Within an architectural examination of suchlike problem formulation, the model has been implemented in the area of architectural layout planning. Referring to what was said before, it is obviously difficult to say which routes architects follow whilst designing. It is therefore complicated to code a set of steps that describe how a design problem could be solved by a machine. Thus, the layout planning problem was set up with a different programming paradigm that specifies a set of constraints which must be met without stating how to achieve this task. A programming language that supports this paradigm and that was used herein is OPL (Optimization Programming Language), which was developed in 1995 [3].

2.1 Programming Techniques

Prototypes have been developed through the use of mathematical (MP), integer (IP), mixed-integer (MIP) and mixed-integer linear programming (MILP) techniques that are based on a geometric or topological description of rooms. The most important advantage of using OPL as a programming language is its ability to specify search procedures as well as upper- and lower-bounds on variables. The use of search procedures is of uppermost interest due to many problems in architectural layout planning being NP-complete. Their application can dramatically influence the total cost of the optimization process.

2.2 Revitalization Models

The revitalization of existing buildings becomes more and more important. We are facing a situation where in many cases there is no need to design new buildings because an increasing number of existing buildings is not being used anymore. The most ecological procedure to revitalize these buildings would be through continuous usage and by making few or no alterations to the stock. Consequentially, the models developed for this purpose are called “destructive” and “non-destructive” models.

3 DESTRUCTIVE MODEL

Destructive Models map the conversion strategy. Large empty spaces within a given floor plan setting are filled with rooms from the room program and thus rearranged. Removal of walls and other structural elements is accepted. This model is mainly applied to buildings with separable primary, secondary, and tertiary structures that were built predominantly using system or skeleton construction methods.

3.1 Methodology

The Destructive Model is based on the inherent structures of an existing building that have arisen from the structural or organizational decomposition of the floor plans. These existent structures are used as a basis for new planning and are applied to the search for solutions. In the Destructive Model, these structures are large empty areas that are filled with the rooms listed in the room program. If these empty areas are not already present, they are created by the architect through decomposition. This means that the application of the Destructive Model is preceded by structural alteration of the floor area to be newly planned. This model aims at arranging the areas defined by the room program within this available space, governed by relationships to be defined by the architect.

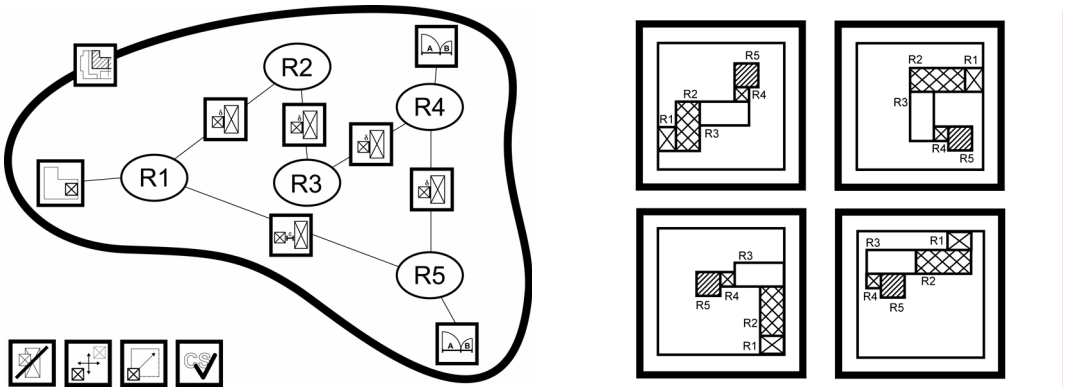


Figure 1. Design Task (Destructive Model) – Alternative Solutions satisfying the same Requirements

3.2 Interdisciplinary Analogies

The methodology of the Destructive Model is based on stepwise partitioning of a building into smaller areas. The subsequent, usually different composition of further areas into a new arrangement is found in various disciplines that employ the methodology of sorting, redistribution or relocation in order to reach a goal. Therefore the model is methodologically analogous to packing problems. In contrast to ordinary packing or bin-packing problems, however, there is no class of objects defined by fixed quantities. Furthermore, the arrangement of rooms is subject to additional conditions (position, proportion, spatial relationships) that render the solution of this problem disproportionably costlier.

3.3 Mathematical Model

The principle of the geometric model adopted is the representation of rooms as rectangular units. Michalek [4] demonstrated this concept in his work on architectural layout planning. In contrast to his concept, a geometric representation was chosen that describes a rectangular unit through a reference point, a length, and a width dimension. Three constraints were taken from this work that describe the location of a unit inside another (Force Inside), the intersection of two units (Prohibit Intersection), as well as the location of a unit on the border

of another unit (Force To Border). Additional constraints were added that specify the connection of two units (Force Connection), the location of a unit on the outside of another unit (Force Outside), as well as the prohibition of a connection between two units (Prohibit Connection). Various design constraints (e.g. aspect ratio, symmetry) that refer to subjective rules were implemented. These design constraints as well as constraint combinations make it possible to extend the architect's ability to intervene in the creative process of automatic layout planning. The use of constraint combinations, for example, led to a new constraint that made it possible to extend the geometric model to non-rectangular units. These so-called Void Units accommodate complex shapes that must not be specified differently from other units, according to their geometrical measures. The mathematical model consists of the following variables, constants and constraints (1-9).

Variables:

Unit i, j	floor space of units
Unit R	floor space of surrounding reference unit
Unit R´	floor space of building site
(I_1, I_2, I_3)	finite-dimensional index sets of rooms
$\varepsilon \geq 0 \;;\; \delta \geq 0$	contact range ; spacing range
$(x_i, y_i) \;;\; (x_j, y_j)$	non-negative reference points unit i,j
$(\Delta x_i, \Delta y_i) \;;\; (\Delta x_j, \Delta y_j)$	non-negative width and length unit i,j

$$u_{ij}, v_{ij} \in \{0,1\} \qquad (u_{ij}, v_{ij}) = \begin{Bmatrix} (0,0), \text{iff. Unit i above Unit j} \\ (0,1), \text{iff. Unit i under Unit j} \\ (1,0), \text{iff. Unit i right of Unit j} \\ (1,1), \text{iff. Unit i left of Unit j} \end{Bmatrix}$$

Variables for Force To Border Constraint:

$$u_{ij}, v_{ij} \in \{0,1\} \qquad (u_{ij}, v_{ij}) = \begin{Bmatrix} (0,0), \text{iff. Unit i on southside} \\ (0,1), \text{iff. Unit i on northside} \\ (1,0), \text{iff. Unit i on westside} \\ (1,1), \text{iff. Unit i on eastside} \end{Bmatrix}$$

Constants:

$(x_R, y_R) \;;\; (x_{R'}, y_{R'}) := (0,0)$	Non-negative reference point units R, R´
$X \;;\; Y \;;\; X' \;;\; Y'$	Non-negative width and length units R, R´

Force Inside Constraint (FInside) (1)

$$x_i \geq x_j \;;\; y_i \geq y_j \;;\; x_i + \Delta x_i - (x_j + \Delta x_j) \leq 0 \;;\; y_i + \Delta y_i - (y_j + \Delta y_j) \leq 0$$

Prohibit Intersection Constraint (PInter) (2)

$$x_i + \Delta x_i - x_j + u_{ij}X' + v_{ij}X' \le 2X'$$
$$x_j + \Delta x_j - x_i + u_{ij}X' - v_{ij}X' \le X'$$
$$y_i + \Delta y_i - y_j - u_{ij}Y' + v_{ij}Y' \le Y'$$
$$y_j + \Delta y_j - y_i - u_{ij}Y' - v_{ij}Y' \le 0$$

Force Connection Constraint (FConn) (3)

$$y_i + \Delta y_i - y_j - u_{ij}Y' + v_{ij}Y' \le Y'$$
$$y_i + \Delta y_i - y_j + u_{ij}Y' - v_{ij}Y' \ge -Y'$$
$$x_i + \Delta x_i - x_j + u_{ij}X' - v_{ij}X' \ge -X' + \delta$$
$$x_j + \Delta x_j - x_i + u_{ij}X' - v_{ij}X \ge -X' + \delta$$

$$y_j + \Delta y_j - y_i - u_{ij}Y' - v_{ij}Y' \le 0$$
$$y_j + \Delta y_j - y_i + u_{ij}Y' + v_{ij}Y' \ge 0$$
$$x_i + \Delta x_i - x_j + u_{ij}X' + v_{ij}X' \ge 0 + \delta$$
$$x_j + \Delta x_j - x_i + u_{ij}X' + v_{ij}X' \ge 0 + \delta$$

$$x_i + \Delta x_i - x_j + u_{ij}X' + v_{ij}X' \le 2X'$$
$$x_i + \Delta x_i - x_j - u_{ij}X' - v_{ij}X' \ge -2X'$$
$$y_i + \Delta y_i - y_j - u_{ij}Y' - v_{ij}Y' \ge -2Y' + \delta$$
$$y_j + \Delta y_j - y_i - u_{ij}Y' - v_{ij}Y' \ge -2Y' + \delta$$

$$x_j + \Delta x_j - x_i + u_{ij}X' - v_{ij}X' \le X'$$
$$x_j + \Delta x_j - x_i - u_{ij}X' + v_{ij}X' \ge -X'$$
$$y_i + \Delta y_i - y_j - u_{ij}Y' + v_{ij}Y' \ge -Y' + \delta$$
$$y_j + \Delta y_j - y_i - u_{ij}Y' + v_{ij}Y' \ge -Y' + \delta$$

Force To Border Constraint (F2Border) (4)

$$x_i + \Delta x_i - (x_j + \Delta x_j) - u_{ij}X' - v_{ij}X' \ge -2X'$$

$$y_i + \Delta y_i - (y_j + \Delta y_j) + u_{ij}Y' - v_{ij}Y' \ge -Y'$$

$$x_i - x_j + u_{ij}X' - v_{ij}X' \le X'$$

$$y_i - y_j - u_{ij}Y' - v_{ij}Y' \le 0$$

Prohibit Connection Constraint (PConn) (5)

$$x_i + \Delta x_i - x_j + u_{ij}(X' + \varepsilon) + v_{ij}(X' + \varepsilon) \le 2X' + \varepsilon$$

$$x_j + \Delta x_j - x_i + u_{ij}(X' + \varepsilon) - v_{ij}(X' + \varepsilon) \le X'$$

$$y_i + \Delta y_i - y_j - u_{ij}(Y' + \varepsilon) + v_{ij}(Y' + \varepsilon) \le Y'$$

$$y_j + \Delta y_j - y_i - u_{ij}(Y' + \varepsilon) - v_{ij}(Y' + \varepsilon) \le -\varepsilon$$

Force Outside Constraint (FOutside) (6)

$$x_i + \Delta x_i - x_j + u_{iR}(X'+\varepsilon) + v_{iR}(X'+\varepsilon) \le 2X'+\varepsilon$$

$$x_j + \Delta x_j - x_i + u_{iR}(X'+\varepsilon) - v_{iR}(X'+\varepsilon) \le X'$$

$$y_i + \Delta y_i - y_j - u_{iR}(Y'+\varepsilon) + v_{iR}(Y'+\varepsilon) \le Y'$$

$$y_j + \Delta y_j - y_i - u_{iR}(Y'+\varepsilon) - v_{iR}(Y'+\varepsilon) \le -\varepsilon$$

$$\varepsilon \ge 0$$

Design Constraints (Length Constraint) (7)

$$\Delta y_i \begin{Bmatrix} \le z_i, \text{ upper boundary of Unit i} \\ = z_i, \text{ exact boundary of Unit i} \\ \ge z_i, \text{ lower boundary of Unit i} \end{Bmatrix}$$

Design Constraints (Area Constraint) (8)

$$\mathrm{F}_i \begin{Bmatrix} \le z_i, \text{ upper area of Unit i} \\ = z_i, \text{ exact area of Unit i} \\ \ge z_i, \text{ lower area of Unit i} \end{Bmatrix}$$

Design Constraints (Perimeter Constraint) (9)

$$U_i \begin{Bmatrix} \le z_i, \text{ upper perimeter of Unit i} \\ = z_i, \text{ exact perimeter of Unit i} \\ \ge z_i, \text{ lower perimeter of Unit i} \end{Bmatrix}$$

$$(z_i \in \mathbb{R}_+ \text{ fixed, } \mathrm{i} \in \{1,...,\mathrm{n}\})$$

4 NON-DESTRUCTIVE MODEL

Non-Destructive models map the reuse strategy, that is, the attempt to reuse the existent room structure of a building while mostly refraining from the removal of walls and other structural elements. The main criterion for this model are the existing structures between room units, for which is determined whether they agree with the structures of a planned future use. This model is mainly applied to buildings with non-separable primary, secondary, and tertiary structures that were built predominantly using massive construction methods.

4.1 Methodology

The areas used by the Non-Destructive Model are assembled from existent rooms. It is possible, albeit not desirable, to apply the model to even smaller units, for example, areas decomposed by a grid. This model aims at finding areas in the existent floor plans that satisfy the requirements of the room plan. This is achieved by comparing the properties of the existing areas with the properties of the areas in the room plan. No structural alterations are performed on the existing building. This is possible because the Non-Destructive Model does not change the geometric shape of a room but merely its use profile.

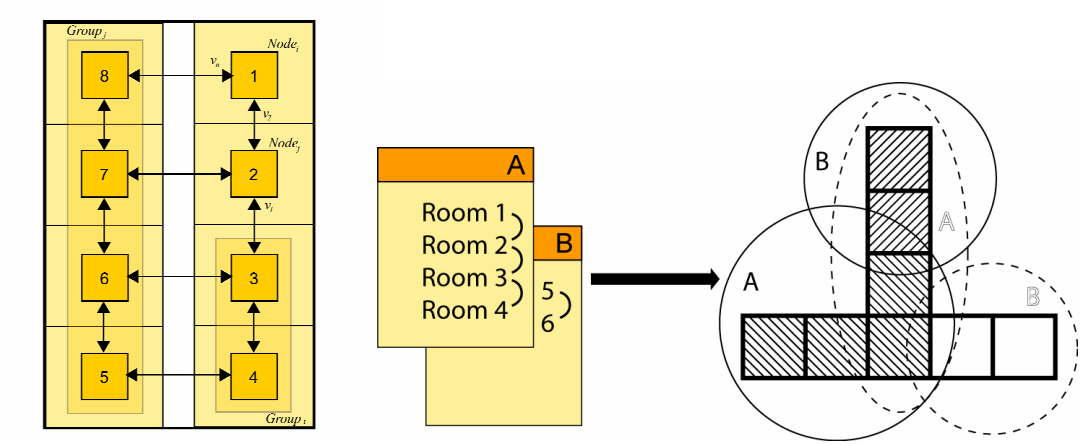


Figure 2. Graph Representation and Methodology of the Non-Destructive Model

4.2 Interdisciplinary Analogies

The methodology of the Non-Destructive Model is similar to the structure of board games such as Chess, Go, or Connect Four. These games are based on a number of fields or points on a grid, which are occupied by pieces according to certain conditions (game rules). The objective is clearly defined, as it is for the Non-Destructive Model. Both share similar features such as a fixed grid, a restricted possibility to occupy fields, and the satisfaction of a higher objective, that is, to win the game.

4.3 Mathematical Model

The Non-Destructive Model focuses on the geometric shapes of existing room limits or the orientation of system lines, such as those of the static system or the grid units of structural components. This abstraction of floor plans ensures non-invasive treatment of existing buildings. Units of the Non-Destructive Model are not limited in their geometric shape. As nodes of a graph they can represent any shape, including non-rectangular shapes. The Non-Destructive Model does not require reference points or distance variables to be determined since the positions of its units are constant.

Conventions:

$G = (V, E)$	graph with
V	set of nodes
$E \subseteq V \times V$	set of edges
$nbreOfNodes$	number of nodes $(= \lvert V \rvert)$
$Node_i$	node i of graph $(i \in \{1,...,\lvert V \rvert\})$
$nbreOfGroups$	number of connected groups (sub graphs) to be found
$Group_j$	group j
$label_{ij}$	means that $label_i = j$

Variables:

$A = a_{ij}$	adjacency matrix of G with $a_{ij} = \begin{Bmatrix} 1, \text{falls } (i,j) \in E \\ 0, sonst \end{Bmatrix}$;
$sizeNodes_i$	size of node i in square meters
$groups_j$	number of required units in group j
$sizeGroups_j$	found total size of group j in square meters
$sizeRooms_{jk}$	size of room k in group j $\left(k \in \{1,\ldots,groups_j\}\right)$
psu	lower limit: percentage of required group size that must at least be met (usually $\leq$ 100%)
pso	upper limit: percentage of required group size that must at most be met (usually $\geq$ 100%)
$label_i$	group to which node i belongs

Number (10)

$$\sum_{i=i}^{|V|} label_{ij} = groups_j \quad \forall j \in \{1,\ldots,nbreOfGroups\}$$

Size (11)

$$\frac{pso}{100} * sizeGroups_j \geq \sum_{i=i}^{|V|} label_{ij} * sizeNodes_i \geq \frac{psu}{100} * sizeGroups_j$$
$$\forall j \in \{1,\ldots,nbreOfGroups\}$$

5 EXAMPLES

5.1 Destructive Model

In this example, a rectangular floor plan with an area of 900 square meters and side lengths of 30 by 30 meters is shown. Satisfying a large number of additional conditions, an arrangement of the 19 areas of the room program had to be found with the sum of the areas of slots 1 through 8 equaling the total area of the building floor plan and with slots 11 through 19 arranged within slot 6.

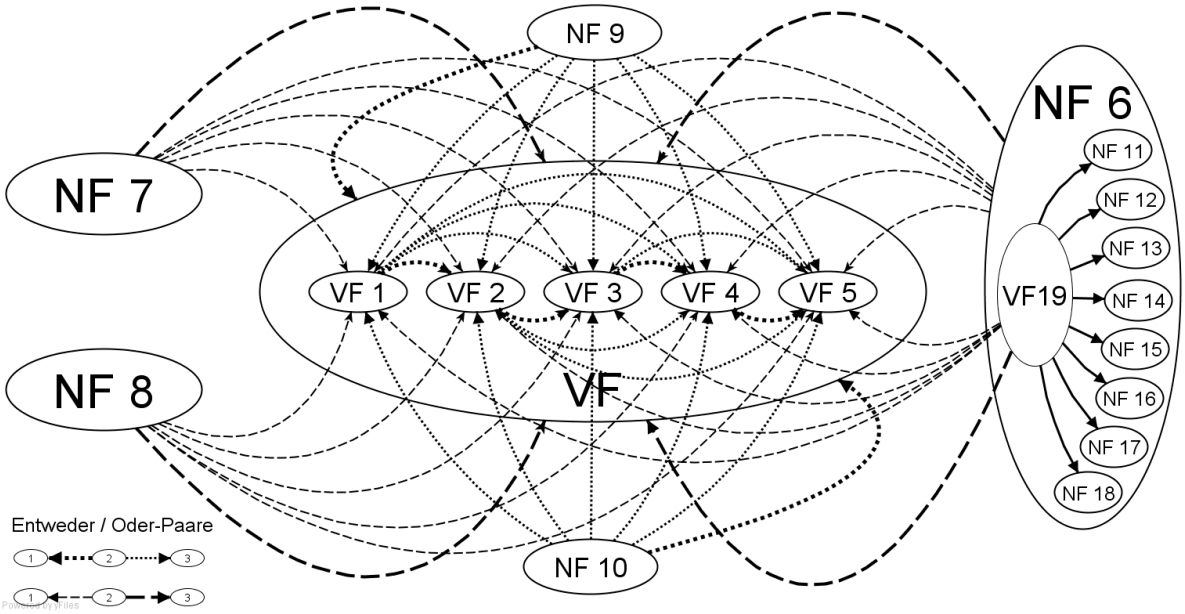


Figure 3. Bubble Diagram showing Constraints

The following figure shows 8 exemplary results of the optimization computation.

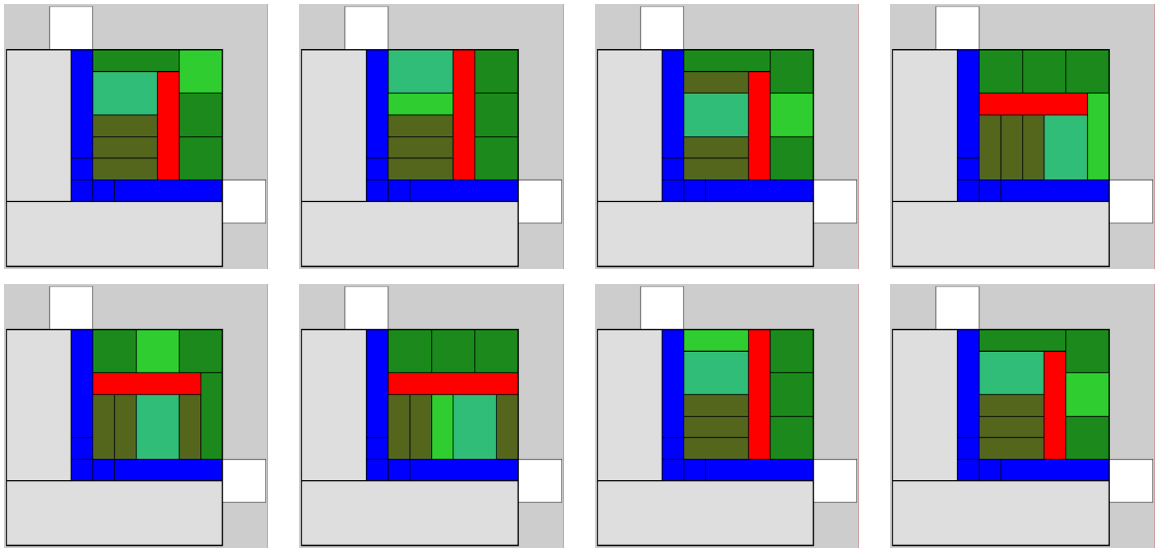
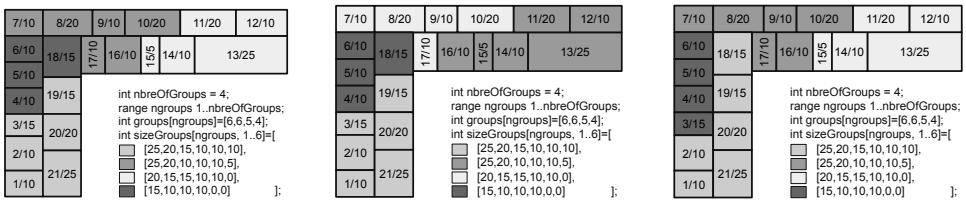


Figure 4. Exemplary Computation Results with fconnectionSet19b

5.2 Non-Destructive Model

The results of the model developed are encouraging. (Figure 5) shows 8 exemplary assemblies of 21 rooms. We were searching for 4 groups of rooms with 6, 6, 5 and 4 group members. All solutions fulfill the requirements made to the size of the rooms, their adjacency, and the number of members in a unit (group).



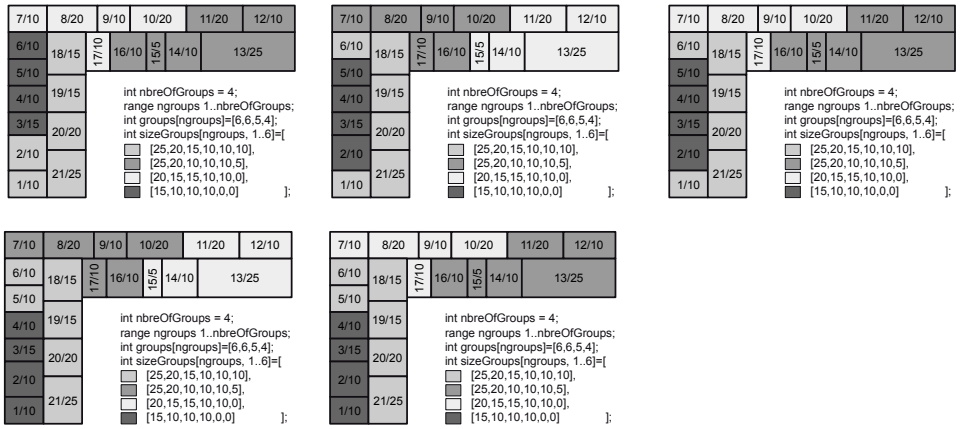


Figure 5. Non-Destructive Optimization of existing Floor Plans

We began working with larger models consisting of 78 rooms, respectively 6,084 entries in the matrix (Figure 6). These optimization runs can be solved within 2 hours on ordinary machines with a reasonable amount of memory. Our latest attempts deal with more than 300 rooms (respectively 90,000 entries in the matrix) arranged on different stories. These tasks are difficult to solve from a computational point of view and might call for parallelization of the programs developed. However, the matrix is capable of representing two- or three-dimensional arrangements (multiple stories) of rooms in equal measure. It can also represent various buildings.

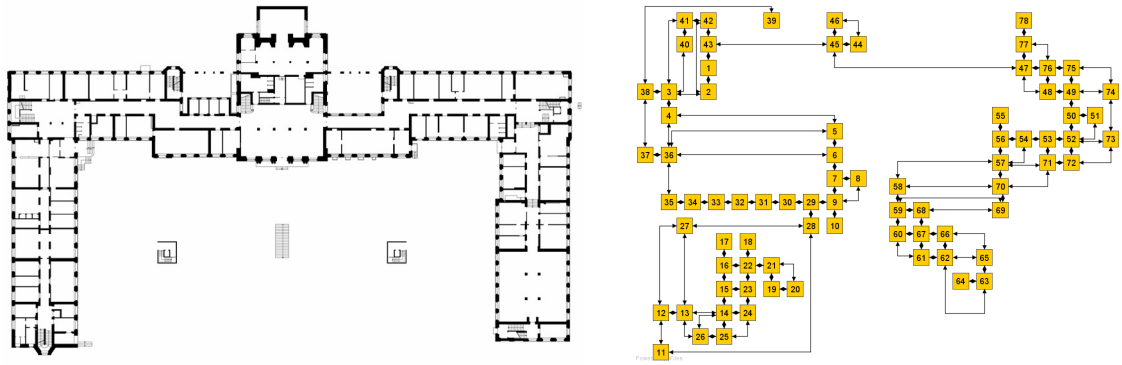


Figure 6. Plan and Graph Representation

(Figure 7) demonstrates an exemplary calculation within which we were searching for 17 rooms. Each room had to fulfill specific requirements regarding its floor space and adjacency to other rooms. The figure represents 3 out of 4,935 solutions found to the problem. All solutions range within a 3% tolerance of the specified objectives. To judge upon these solutions, we integrated quality ratings and performance measures as well as penalizations into the model. (Figure 8)

Figure 7. Three Exemplary Solutions

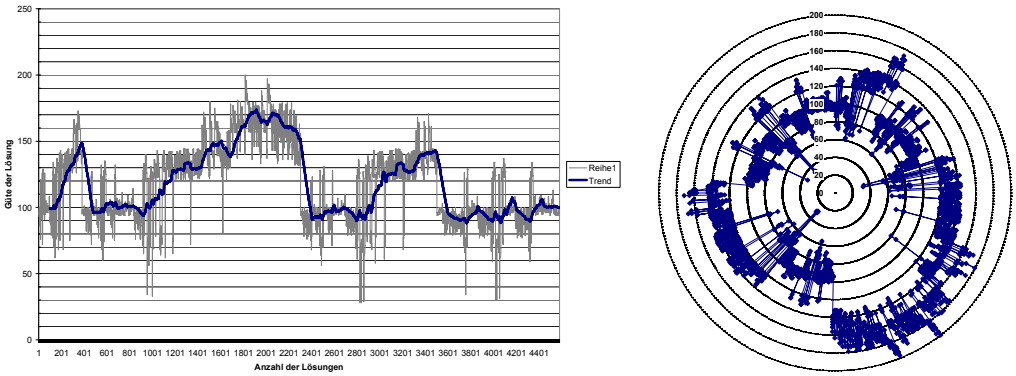


Figure 8. Calculation Trend and Performance Measure

On the basis of these performance measures, a solution can objectively be evaluated. (Figure 9) shows the solution with the best performance. The numbers demonstrate a nominal / actual value comparison of the floor spaces we were searching for.

Figure 9. Best Performance and nominal / actual Value Comparison

6 CONCLUSION

Calculations performed using Destructive and Non-Destructive Models assert that they can be used to solve revitalization problems in floor plan design. Normative influences and design goals can be defined as additional conditions and target functions, which are included in the optimization computation. The results are very promising, especially in terms of producing a large number of different solutions that all satisfy the same requirements. This large number of producible solutions is a testimony to the potential application of this tool: The computation of a large number of perfect solutions for a problem, the automated evaluation of its performance, and the manual comparison of the best solutions by the architect, who then continues his planning activities on the basis of these solutions.

REFERENCES

[1] Domschke, W. and A. Drexl (2005). *Einführung in Operations Research* : mit 63 Tabellen. Berlin [u.a.], Springer.

[2] Bhatti, M. A. (2000). *Practical optimization methods: with mathematica applications.* New York, NY [u.a.], Springer.

[3] VanHentenryck, P. and I. Lustig (1999). *The OPL optimization programming language.* Cambridge, Mass [u.a.], MIT Press.

[4] Michalek, J. J. (2001). *Interactive Layout Design Optimization.* Optimal Design Laboratory. Michigan, University of Michigan**:** 115.